

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

VYTVOŘENÍ 3D MODELU Z VÍCE POHLEDŮ KAMER

3D MODEL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Sejkora

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miloslav Richter, Ph.D.

BRNO 2019

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Martin Sejkora

ID: 195429

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Vytvoření 3D modelu z více pohledů kamer

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte pracoviště pro vytvoření prostorového modelu objektu (bez spodní části) za pomoci několika kamer.

Realizujte pro objekty, které mají samy o sobě dostatek bodů pro rekonstrukci.

1) Nastudujte metody prostorové rekonstrukce objektu za pomoci soustavy kamer.

2) Definujte požadavky na měřicí systém v závislosti na tvaru a velikosti objektu. Zvolte vhodnou kategorii objektů a pro ni realizujte měřicí pracoviště.

3) Nasnímejte sady snímků pro testování metod rekonstrukce.

4) Realizujte vhodné metody rekonstrukce a zhodnoťte kvalitu dosažených výsledků. Výsledky uložte ve vhodném datovém formátu.

DOPORUČENÁ LITERATURA:

Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 2005, ISBN 978-80-2-1-0454-5

Sonka M., Hlavac V., Boyle R.: Image Processing, Analysis, and Machine Vision, 3rd Edition, CL Engineering, 2007, ISBN: 978-0495082521

Termín zadání: 4.2.2019

Termín odevzdání: 20.5.2019

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík,
CSc.

předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Brno

Abstrakt

Tato práce pojednává o problematice počítačového vidění a o problematice zpětné prostorové rekonstrukce objektu ze snímků kamer. Popisuje teoretické základy potřebné k pochopení problematiky a postup možného řešení. Softwarové řešení práce je vytvořeno v jazyce C++ a využívá knihovnu OpenCV.

Klíčová slova

kamera, snímání, stereo, openCV, rekonstrukce, model

Abstract

This document focuses on computer vision and backward spatial reconstruction of the object from camera images. It describes the theoretical foundations needed to understand the problem and how to solve it. Software solution is created in C++ language and uses OpenCV library.

Keywords

camera, scanning, stereo, openCV, reconstruction, model

Bibliografická citace:

SEJKORA, Martin. *Vytvoření 3D modelu z více pohledů kamer*. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/119314>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. Miloslav Richter, PhD.

Prohlášení autora o původnosti díla

„Prohlašuji, že svou bakalářskou práci na téma Vytvoření 3D modelu z více pohledů kamer jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.“

V Brně dne: **20. května 2018**

.....
Martin Sejkora, podpis

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Miloslavu Richterovi, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc při zpracování mé bakalářské práce.

V Brně dne: **20. května 2019**

.....
Martin Sejkora, podpis

Obsah

1	Úvod.....	9
2	Kamera	10
2.1	Matematický model kamery	10
2.2	Reálná kamera	11
2.3	Zkreslení obrazu	12
2.4	Vnější parametry kamery	12
2.5	Projekční matice	13
3	Stereoskopie	14
3.1	Epipolární geometrie	14
3.2	Fundamentální matice	15
3.3	Triangulace	15
4	Detekce významných bodů	17
4.1	Harrisův detektor	17
4.2	SIFT detektor	18
4.3	SURF detektor	18
5	Vlastní řešení.....	19
5.1	Využité prostředky	19
5.1.1	Knihovna OpenCV	19
5.1.2	Kamera DFK 41BU02.....	19
5.2	Kalibrace kamery.....	20
5.3	Pozice kamery	21
5.4	Detekce a korespondence bodů	22
5.5	Výpočet prostorových souřadnic	24
5.6	Vizualizace získaných dat	25
5.7	Měřicí pracoviště	25
5.8	Popis programů.....	27
5.8.1	Program pro kalibraci kamery – CameraCalibration	27
5.8.2	Program pro vytvoření modelu – 3DModel	28
6	Závěr	31

Seznam obrázků

Obrázek 1: Dírková kamera [4]	10
Obrázek 2: Ekvivalentní model dírkové kamery [4]	11
Obrázek 3: Typy zkreslení [3]	12
Obrázek 4: Epipolární geometrie [3]	14
Obrázek 5: Triangulace [3]	15
Obrázek 6: Princip Harrisova detektoru [8]	17
Obrázek 7: Závislost detekce rohu na zvětšení [9]	18
Obrázek 8: Kamera DFK 41BU02	20
Obrázek 9: Šachovnicový vzor o velikosti 9x6	20
Obrázek 10: Šachovnice s detekovanými rohy	21
Obrázek 11: Značka Aruco 4x4 se znázorněnou mřížkou	21
Obrázek 12: Detekce značek Aruco	22
Obrázek 13: Detekce významných bodů pomocí SURF	23
Obrázek 14: Nalezené korespondence	24
Obrázek 15: Návrh měřicího pracoviště	26
Obrázek 16: Měřicí pracoviště	27
Obrázek 17: Postup hledání korespondencí	29
Obrázek 18: Ukázka modelu krychle	30

1 ÚVOD

Účelem této bakalářské práce je přiblížit problematiku získávání prostorové informace o objektech ze snímků kamer. Cílem bude prozkoumat možnosti prostorové rekonstrukce a sestavit prostorový model snímaného objektu.

Existuje více metod pro prostorovou rekonstrukci objektu, v této práci se bude pojednávat o rekonstrukci z dvourozměrných fotografií bez dalších pomocných zařízení (např. laser či jiné nasvícení objektu, vzdálenostní čidla atd.). Jedná se o odvětví výpočetní techniky zvané počítačové vidění (z angl. computer vision).

Měřicí systém bude určen pro objekty do velikosti krychle o hraně 40 centimetrů. Bude se jednat o dostatečně texturované objekty, aby bylo možné detekovat dostatek významných bodů pro rekonstrukci. Pro implementaci dále uvedených metod bude použita knihovna pro práci s obrazem OpenCV.

Práce bude rozdělena na teoretickou a praktickou část. Teoretická část se skládá ze tří kapitol. První kapitola pojednává o kameře, jejím matematickém modelu, popisu reálné kamery a způsobu, jak se bod promítá do obrazu. Druhá kapitola přibližuje problematiku stereoskopického snímání a matematicky tento způsob snímání popisuje. Třetí kapitola obsahuje informace o detekci významných bodů v obraze.

Praktickou část této práce zapouzdřuje jedna hlavní kapitola, v které jsou postupně uvedeny použité prostředky pro práci, jednotlivé postupy potřebné pro prostorovou rekonstrukci objektu, popisuje měřicí pracoviště vytvořené pro tuto práci a popisuje programy, které byly v rámci práce vytvořeny.

V závěru se poté zhodnotí dosažené výsledky této práce.

2 KAMERA

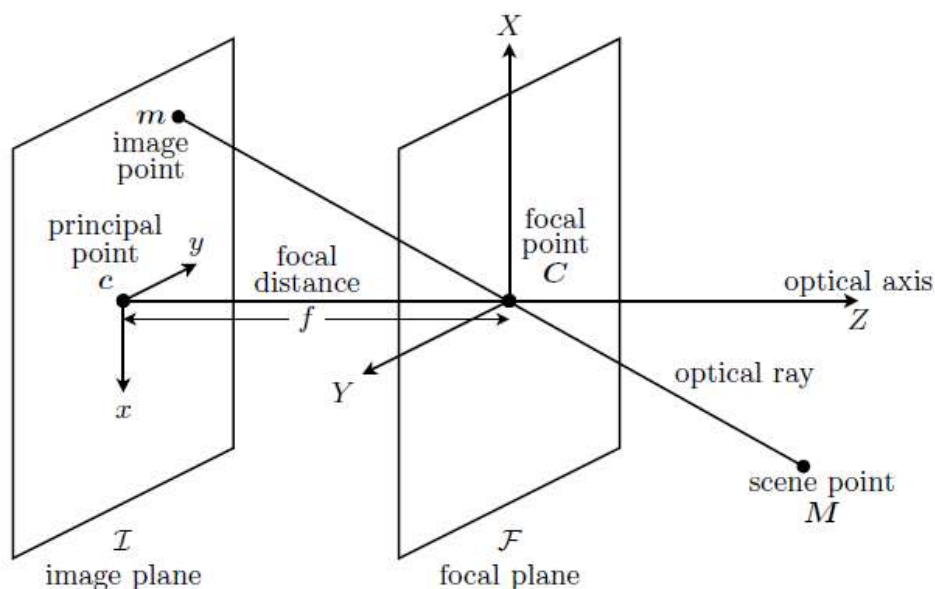
Tato kapitola se zabývá popisem kamery a způsobem, jakým se scéna (reprezentovaná pro jednoduchost jedním bodem) promítá do obrazu. Všechny postupy jsou matematicky popsány a budou využívány jako základ pro rekonstrukci scény.

2.1 Matematický model kamery

Nejjednodušším modelem kamery je tzv. dírková kamera (anglicky pinhole camera). Princip kamery je znázorněn na obrázku 1. Světlo dopadající na projekční desku (image plane) prochází malým otvorem – středem projekce C , který leží na optické ose (optical axis). Na jeden bod desky dopadá paprsek světla vedený z reálného bodu M přes střed projekce. Bod dopadu paprsku označíme jako obraz bodu M . Velikost získaného obrazu je závislá na ohniskové vzdálenosti (focal distance). Z podobnosti trojúhelníků dostaneme rovnici:

$$\frac{-m}{f} = \frac{M}{Z} \quad (1)$$

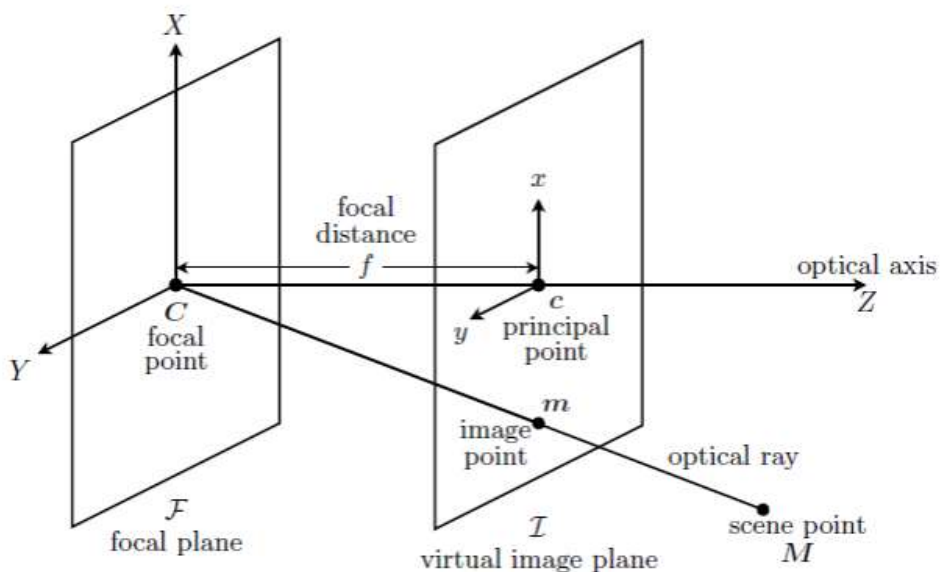
kde m je velikost obrazu, M velikost objektu, f je ohnisková vzdálenost a Z je vzdálenost objektu.



Obrázek 1: Dírková kamera [4]

Abychom se vyhnuli převrácenému obrazu, použijeme ekvivalentní model dírkové kamery (Obrázek 2). Tento model se oproti standartní dírkové kameře liší v tom, že projekční deska leží mezi objektem a středem projekce. To způsobí, že výsledný obraz nebude převrácený. Výsledná rovnice je opět dána podobností trojúhelníků, avšak již neobsahuje záporné znaménko.

$$\frac{m}{f} = \frac{M}{Z} \quad (2)$$



Obrázek 2: Ekvivalentní model dírkové kamery [4]

Pokud reálnému bodu M přiřadíme prostorové souřadnice $[x, y, z]$ a bodu obrazu m přiřadíme souřadnice $[u, v]$, pak můžeme promítnutí bodu na projekční rovinu vyjádřit pomocí rovnic:

$$u = f \cdot \frac{x}{z}; \quad v = f \cdot \frac{y}{z} \quad (3)$$

kde f je ohnisková vzdálenost. Po přepsání do maticového zápisu pak dostáváme rovnici:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4)$$

Uvedené rovnice ovšem platí pouze v ideálním matematickém modelu.

2.2 Reálná kamera

Jelikož se optický snímač kamery nenachází přesně na optické ose, musíme zavést parametry pro vyjádření posunutí senzoru, které vyjadřují rozdíl polohy středu souřadného systému optického snímače a středu souřadného systému projekční roviny. Dále musíme zavést rozdílné ohniskové vzdálenosti pro směry osy x a y . To že se ohniskové vzdálenosti uvádějí pro osu x i y je důsledek faktu, že pixely na snímacích senzorech nemají přesně čtvercový tvar. Po zavedení těchto parametrů dostáváme upravené rovnice pro promítnutí bodu na projekční rovinu:

$$u = f_x \frac{x}{z} + c_x; \quad v = f_y \frac{y}{z} + c_y \quad (5)$$

kde c_x a c_y vyjadřují posunutí senzoru pro dané směry a f_x a f_y jsou ohniskové vzdálenosti. V maticovém tvaru pak dostáváme rovnici:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6)$$

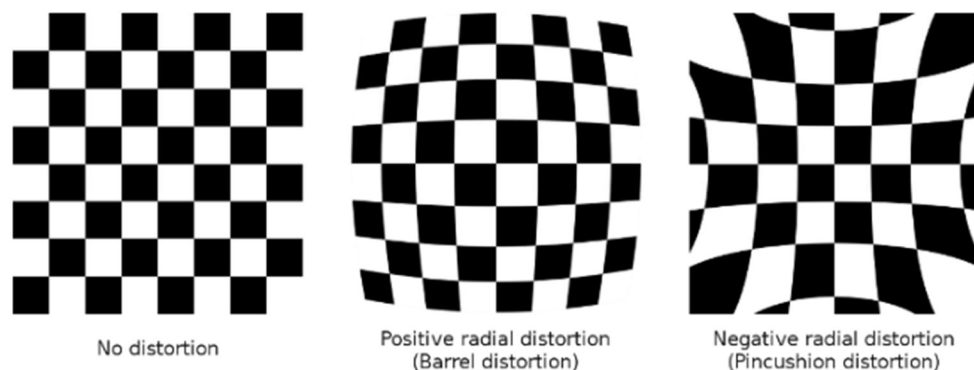
Matici 3x3 označujeme jako matici kamery nebo matici vnitřních parametrů. V této práci bude označena C .

$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

2.3 Zkreslení obrazu

Reálné kamery vykazují řadu optických vad, které způsobí zkreslení (anglicky distortion). Zkreslení může být způsobeno špatným umístěním čoček optické soustavy, nevhodné zaostření, pohyb snímáče a předmětu, anebo je způsobeno nedokonalostí čoček. Zkreslení se projevuje „ohnutím“ rovných linií v obraze. Nejvíce znatelné je u okrajů obrazu.

Zkreslení existuje celá řada, ovšem základní je radiální zkreslení, které můžeme rozdělit na pozitivní a negativní, resp. na soudkovité zkreslení (barrel distortion), kde jsou vnitřní body vzoru zvětšeny více než vnější body, a zkreslení poduškovité (pincushion distortion), kde jsou vnější body zvětšeny více než vnitřní body [6].



Obrázek 3: Typy zkreslení [3]

2.4 Vnější parametry kamery

Jako vnější parametry kamery označujeme proměnné, které nám určují pozici a natočení kamery v prostoru v závislosti na souřadnicích scény, které nazýváme světové souřadnice. Tyto parametry můžeme rozdělit na parametry pozice a natočení. Pozici kamery určuje translační vektor t .

$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (8)$$

Tento vektor může být interpretován jako pozice počátku světového souřadnicového systému v souřadnicích kamery.

Natočení kamery pak určuje matice rotace R .

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (9)$$

Řádky matice reprezentují směr os světového souřadnicového systému v souřadnicích kamery.

Výše uvedené parametry můžeme sloučit do tzv. matice vnějších parametrů (extrinsic camera matrix), která obsahuje matici rotace v levém bloku a translační vektor v pravém bloku [5].

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (10)$$

2.5 Projekční matice

Projekční matice nám určuje, jak se promítne reálný bod ve světovém souřadnicovém systému do souřadnic obrazu kamery. K jejímu vytvoření nám poslouží matice vnitřních parametrů (7) a matice vnějších parametrů (10). Projekční matici označíme jako **P**.

$$P = C[R|t] \quad (11)$$

Nyní můžeme pomocí této matice vyjádřit transformaci souřadnic bodu v třírozměrném prostoru do souřadnic obrazu, a to podle rovnice:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = C[R|t] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (12)$$

Bod X se tedy v obraze objeví jako bod x podle rovnice:

$$x = P \begin{bmatrix} X \\ 1 \end{bmatrix} \quad (13)$$

3 STEREOSKOPIE

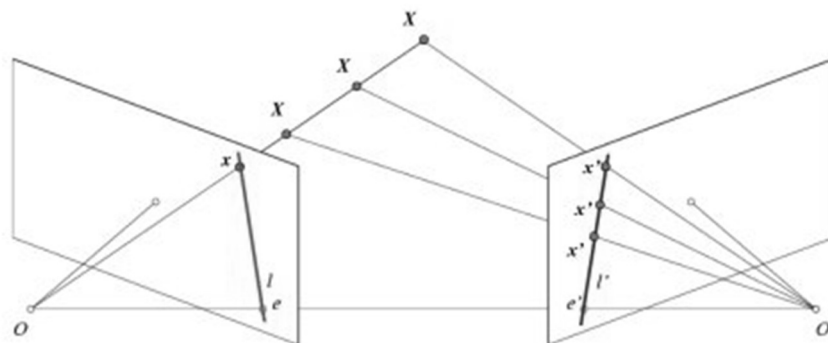
V následující kapitole se budeme zabývat snímáním scény pomocí více kamer. Seznámíme se s principy snímání, geometrickým popisem pohledů dvou kamer a s matematickým popisem získávání informace o hloubce obrazu.

Pokud ke snímání scény použijeme pouze jednu kameru, ztrácíme důležitou informaci o hloubce scény, jelikož transformujeme třírozměrný objekt do dvourozměrného obrazu. Řešením tohoto problému je použití více kamer. Pokud danou scénu budeme snímat pomocí dvojice kamer, hovoříme o stereoskopickém snímání. Tento způsob snímání funguje podobně jako lidské oči. Mozek ze dvou navzájem posunutých obrazů získá informaci o vzdálenosti objektů. Stejným způsobem budeme získávat trojrozměrnou rekonstrukci měřeného objektu.

Při stereoskopickém snímání musí být kamery vzájemně posunuté a jejich zorná pole se musejí překrývat. Stereoskopické snímání nemusí nutně znamenat pohled právě dvou kamer, ale může být reprezentováno jednou kamerou, kterou postupně snímáme scénu či objekt z různých umístění. V tomto případě je nutné zajistit, aby scéna byla statická.

3.1 Epipolární geometrie

Jako epipolární geometrii označujeme geometrii mezi dvěma pohledy kamer. Způsob snímání jedné scény dvěma kamerami je znázorněn na obrázku č. 4.



Obrázek 4: Epipolární geometrie [3]

Pokud ke snímání použijeme pouze levou kameru, nejsme schopni určit bod v prostoru, který by odpovídal bodu x v obraze, jelikož každý bod na úsečce OX se promítá do stejného bodu. Pokud vezmeme v potaz i pravý snímek, vidíme, že se různé body na úsečce OX promítají jako různé body x' . Tudíž jsme schopni pomocí těchto dvou obrazů určit správný bod v prostoru.

Projekce různých bodů na úsečce OX nám v pravém obraze vytváří přímku l' . Tato přímka se nazývá epipolára. To znamená, že bod x z levého obrazu budeme v pravém obraze hledat právě na epipoláře. Toho se dá využít při hledání korespondujících bodů. Rovinu XOO' nazýváme epipolární rovina. Body O a O' jsou středy projekce kamer. Jejich projekce e a e' označujeme jako epipóly [3].

3.2 Fundamentální matice

Fundamentální matice (angl. fundamental matrix) je matematickým vyjádřením epipolární geometrie. Popisuje vztah mezi bodem v obraze jedné kamery a jeho epipolárou v obraze druhé kamery. Opět budeme vycházet z obrázku č. 4. Hledáme tedy vztah mezi bodem x a epipolární přímkou l' definovaný fundamentální maticí F o rozměrech 3×3 , který můžeme vyjádřit rovnicí [7]:

$$l' = Fx \quad (14)$$

Vztah mezi dvěma korespondujícími body můžeme popsat rovnicí [7]:

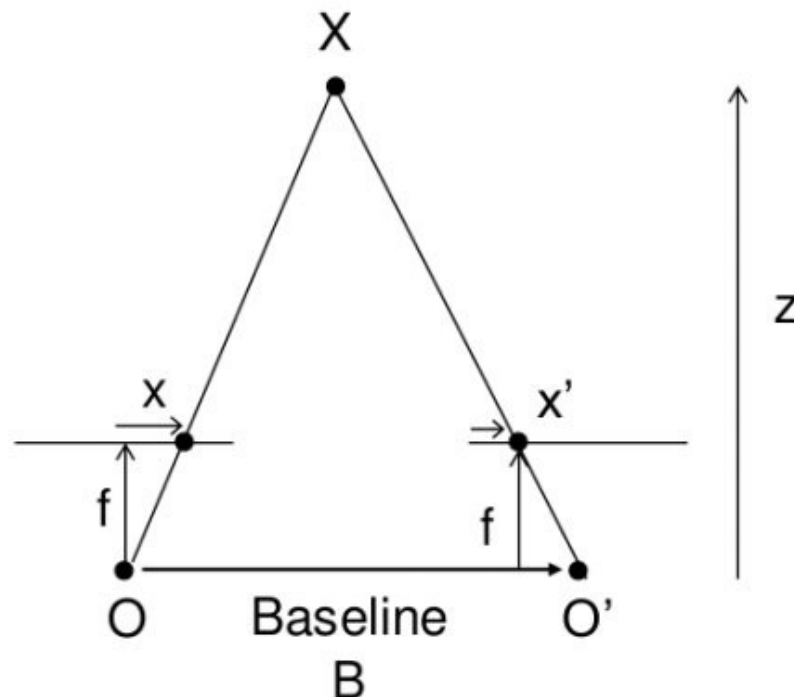
$$x'^T F x = 0 \quad (15)$$

Fundamentální matici můžeme stanovit pomocí znalosti korespondujících bodů. Rozšířenou metodou je tzv. osmibodový algoritmus (8-point algorithm), který stanoví fundamentální matici na základě osmi korespondencí [3]. Dále lze získat fundamentální matici z projekčních matic kamer.

Speciálním případem fundamentální matice je esenciální matice E (essential matrix). Tato matice popisuje stejně jako fundamentální matice vztah mezi body ve dvou pohledech kamer, ale neobsahuje žádné informace o kamerách samotných (tzn. kalibrační matici a koeficienty zkreslení). V případě ideální kamery jsou tyto matice stejné ($E = F$).

3.3 Triangulace

Triangulace je metoda pro získávání informace o vzdálenosti snímaného bodu. Opět využívá dvou snímků téže scény. Je založena na geometrických vlastnostech trojúhelníku. Princip triangulace si vysvětlíme pomocí obrázku č. 5.



Obrázek 5: Triangulace [3]

Bod X je zde snímán dvěma kamerami se středy projekce O a O' a je promítán do obrazů jako x a x'. B je vzdálenost mezi kamerami a f je ohnisková vzdálenost (oba dva parametry známe) [3]. Z podobnosti trojúhelníků OO'X a xx'X můžeme sestavit rovnici:

$$\frac{B - (x - x')}{Z - f} = \frac{c}{f} \quad (16)$$

Z rovnice (16) můžeme vyjádřit Z, což je vzdálenost bodu od kamery:

$$Z = \frac{Bf}{x - x'} \quad (17)$$

Rozdíl souřadnic bodů x a x' nazýváme disparita. Z rovnice (17) tedy vyplývá, že vzdálenost bodu je nepřímo úměrná disparitě.

Pomocí výše uvedených rovnic jsme tedy schopni získat informaci o vzdálenosti bodu od kamery. Pokud tyto znalosti spojíme se znalostí projekční matice kamery, resp. znalosti pozice kamery v prostoru a jejím natočení, jsme schopni získat pozici daného bodu ve světových souřadnicích. Můžeme tudíž zpětně zrekonstruovat pozici bodů v prostoru.

4 DETEKCE VÝZNAMNÝCH BODŮ

Nejdůležitějším problémem 3D rekonstrukce pomocí pohledů kamer je bezesporu detekce významných bodů v obraze, které posléze podrobíme výše uvedeným metodám, a tak získáme informaci o jejich pozici v prostoru. Jelikož tato práce řeší rekonstrukci pouze ze snímků kamer, bude výsledný objekt tvořen pouze detekovanými body. Tato kapitola se zaměří na detektory významných bodů a na další problémy ohledně detekce v obraze.

Významný bod je takový bod v obraze, který vykazuje vysoký gradient jasové funkce v definovaném okolí. To můžeme chápat tak, že je to bod, který je nejméně podobný svému blízkému okolí (jako jsou vrcholy, rohy, hranice)[8].

4.1 Harrisův detektor

Tento rohový detektor byl poprvé publikovaný v roce 1988 (Chris Harris, Mike Stephens: A Combined Corner and Edge Detector). Tato metoda hledá výrazné jasové změny ve všech směrech v místě významného bodu. Tyto změny hledá posuvem lokálního okénka. Pro potlačení šumu se volí okénko s Gaussovým vyvážením hodnot.



Obrázek 6: Princip Harrisova detektoru [8]

Algoritmus hledání bodu je založen na lokální autokorelační funkci:

$$c(x, y) = \sum_{(u,v) \in W(x,y)} w(u, v) \cdot [I(u, v) - I(u + \Delta x, v + \Delta y)]^2 \quad (18)$$

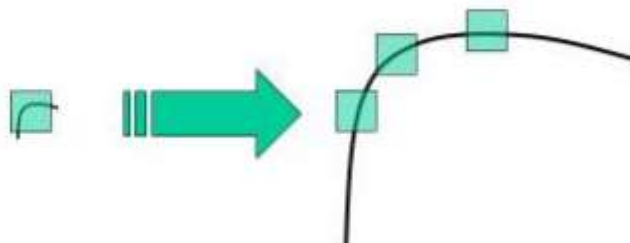
kde $W(x,y)$ je vyšetřované okénko na pozici x,y , (u,v) jsou souřadnice v rámci okénka, $w(u,v)$ je váhová funkce (zde Gaussova funkce), $I(u,v)$ je jasová funkce a $(\Delta x, \Delta y)$ je lokální posun okénka.

Dále je nutné provést další výpočetní úkony, jejichž vyjádření můžeme nalézt v literatuře (např. v [8]). Ve stručnosti uvedu postup výpočtů: nejprve vypočteme funkce I_x a I_y pomocí Taylorova rozvoje, poté vypočítáme prvky autokorelační matice C , a nakonec rozhodneme, zda je daný bod významný pomocí funkce H (indikátor přítomnosti rohu):

$$H(\lambda_1, \lambda_2) = \lambda_1 \cdot \lambda_2 - \kappa \cdot (\lambda_1 + \lambda_2) \quad (19)$$

kde λ_1 a λ_2 jsou vlastní čísla matice C a κ je konstanta (volí se od 0,04 do 0,15)[8].

Výhody tohoto detektoru jsou nezávislost na posunutí a rotaci, nezávislost na změnu intenzity jasu a nízká výpočetní náročnost. Nevýhodou je závislost na změně měřítka. Pokud tedy aplikujeme Harrisův detektor na jeden rozdílně zvětšený roh, nebude jeho odezva vždy stejná (znázorněno na obrázku č. 7).



Obrázek 7: Závislost detekce rohu na zvětšení [9]

4.2 SIFT detektor

Jak bylo uvedeno v předchozí kapitole, nevýhodou Harrisova detektoru je závislost na měřítku. Tento problém řeší detektor SIFT (Scale-Invariant Feature Transform), který je nezávislý na rozdílném zvětšení snímku (je invariantní na měřítko). Navíc kromě hledání významných bodů určuje také jejich deskriptory. Deskriptor slouží jako popis významného bodu, resp. popisuje jeho nejbližší okolí. Pomocí deskriptorů můžeme hledat dvojice korespondujících bodů ze dvou snímků téže scény.

Popis algoritmu SIFT:

1. Detekce kandidátů na významné body
2. Vyřazení nestabilních bodů (ponechají se pouze rohy), určení polohy bodů
3. Určení orientace významných bodů
4. Výpočet příznakového vektoru pro každý bod

První dva body algoritmu slouží k určení významného bodu, druhé dva pak k přiřazení deskriptoru. [9].

Detektor SIFT je invariantní vůči rotaci, posunu, změně jasu a kontrastu a změně měřítka, ovšem je výpočetně velmi náročný.

4.3 SURF detektor

SURF detektor, stejně jako SIFT, detekuje významné body a přiřazuje jim deskriptory. Zkratka SURF znamená Speed-Up Robust Features. To napovídá, že hlavním cílem SURF je snížení výpočetní náročnosti při zachování výkonnosti.

Algoritmus SURF můžeme definovat čtyřmi kroky:

1. Detekce významných bodů
2. Určení přesné polohy významných bodů
3. Určení orientace významných bodů
4. Popis jednotlivých významných bodů

Jak můžeme vidět, algoritmus je velmi podobný s algoritmem SIFT, používá ovšem odlišné procedury v jednotlivých krocích (kvůli snížení výpočetní náročnosti)[9].

5 VLASTNÍ ŘEŠENÍ

V následující části práce budou popsány postupy, kterých bude využito pro získání prostorového modelu objektu. Seznámíme se s jednotlivými kroky, které jsou k získání modelu nutné, a také s měřicím pracovištěm. Budou také přesně popsány programy vytvořené v rámci této práce.

5.1 Využité prostředky

5.1.1 Knihovna OpenCV

OpenCV (Open Source Computer Vision Library) je soubor knihoven pro práci s počítačovým viděním. Byla vytvořena tak, aby poskytovala společnou infrastrukturu pro aplikace počítačového vidění a urychlila používání strojového vnímání. Jelikož jde o produkt s licencí BSD, lze tuto knihovnu zdarma využívat pro akademické a komerční účely.

Knihovna má více než 2500 optimalizovaných algoritmů, které mohou být použity k detekci a rozpoznání tváří, identifikaci objektů, sledování pohybu kamer, sledování pohybujících se objektů, vytváření 3D modelů ze stereofonních kamer, najít podobné snímky z obrazové databáze, odstranit červené oči ze snímků pořízených pomocí blesku, sledovat pohyby očí atd. Knihovna má rozhraní C++, C, Python, Java a Matlab. Podporuje operační systémy Windows, Linux, Mac OS a Android.[2]

V této práci budou veškeré popsané postupy a algoritmy implementovány pomocí funkcí knihovny OpenCV. Proto budou u jednotlivých kapitol uvedeny použité funkce (v jazyce C++) a bude vysvětlena jejich funkčnost.

5.1.2 Kamera DFK 41BU02

Pro účely této práce jsem použil kameru DFK 41BU02 od firmy Imaging Source. Jedná se o barevnou průmyslovou kameru. Využívá ½palcový Sony CCD senzor. Její rozlišení je 1280x960 pixelů a zvládne až 15 snímků za sekundu. Na kameru je připevněn objektiv s ohniskovou vzdáleností 8 mm a clonovým číslem 1,2, který má ruční ostření a ovládání clony. Pro připojení kamery je využito rozhraní USB 2.0 a snímky jsou získávány pomocí softwaru IC Capture 2.4.

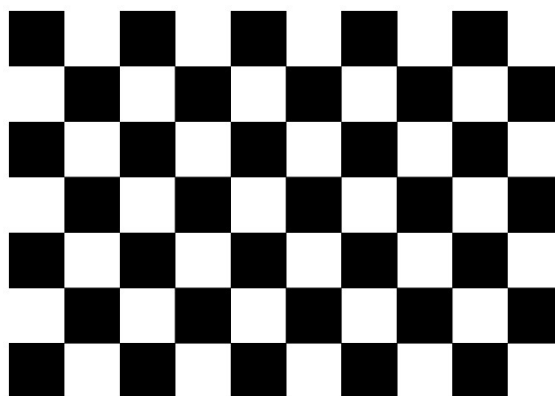


Obrázek 8: Kamera DFK 41BU02

5.2 Kalibrace kamery

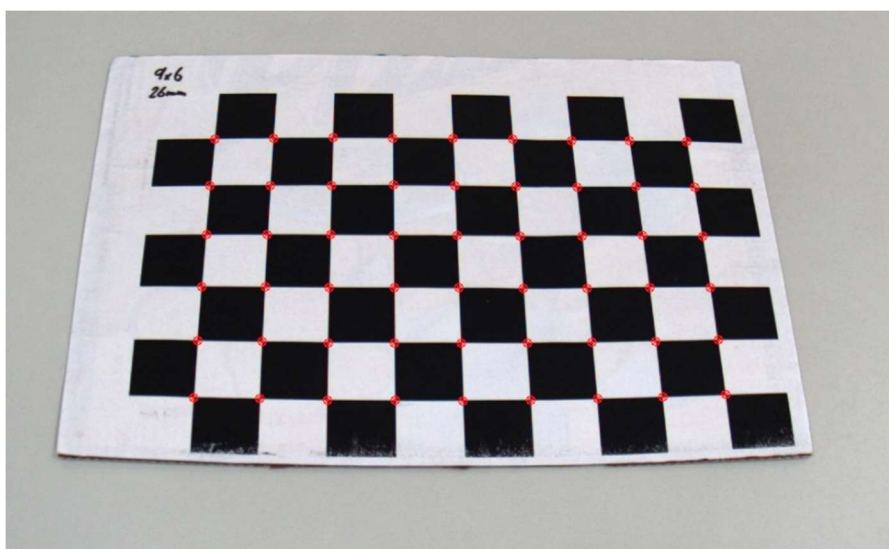
Kalibrací kamery nazýváme proces zjištění vnitřních parametrů kamery, tedy matice vnitřních parametrů, která zapouzdřuje ohniskové vzdálenosti a posunutí senzoru (viz kapitola 2.2), a dále také koeficienty zkreslení.

Základem kalibrace je nasnímáním kalibračního vzoru, který představuje černobílá šachovnice se známými vlastnosti, což je počet polí (resp. počet rohů) a jejich rozměr. Tento vzor je nutné nasnímat z různých úhlů a pohledů tak, aby šachovnicový vzor zabíral podstatnou část snímku. Na těchto snímcích se poté rozpoznají rohy šachovnice. Výsledné parametry se vypočítají ze zjištěných pozic rozpoznaných rohů. Přesnost měření bude vyšší s rostoucím počtem kalibračních snímků.



Obrázek 9: Šachovnicový vzor o velikosti 9x6

Pro rozpoznání rohů šachovnice je využita funkce `cv::findChessboardCorners`, která na vstupních snímcích rozezná rohy šachovnice a předá jejich pozici ve vektoru. Nalezené body je možné zobrazit pomocí funkce `cv::drawChessboardCorners` (viz obrázek č. 10). Poté jsou tyto body společně s vektorem prostorových bodů a rozměry obrazu předány funkci `cv::calibrateCamera`. Tato funkce nám ze zadaných hodnot vypočítá matici vnitřních parametrů a koeficienty zkreslení. Tyto informace si můžeme uložit a poté s nimi dále pracovat.



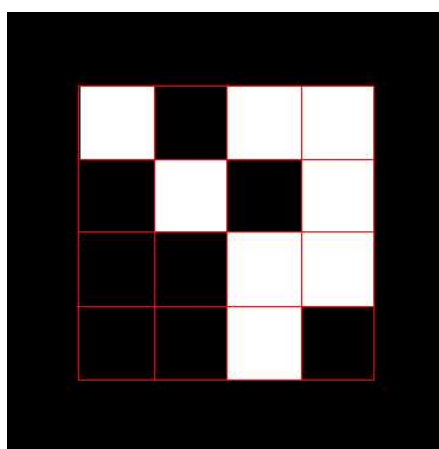
Obrázek 10: Šachovnice s detekovanými rohy

5.3 Pozice kamery

Abychom byli schopni realizovat 3D rekonstrukci objektu, musíme znát přesnou pozici kamery ve světovém souřadném systému. Tuto pozici nám vyjadřuje vektor rotace a vektor translace (posunutí), tedy parametry obsažené v projekční matici. Cílem tedy bude nalezení projekční matice.

Pro zjištění vektorů translace a rotace je nutné určitým bodům ve snímku (pixelovým souřadnicím) přiřadit souřadnice světového souřadného systému. Je tedy nutné umístit do scény objekt, na kterém dokážeme rozpoznat body a těmto bodům přiřadit světové souřadnice. Pro tuto úlohu jsem se rozhodl využít značek z knihovny Aruco.

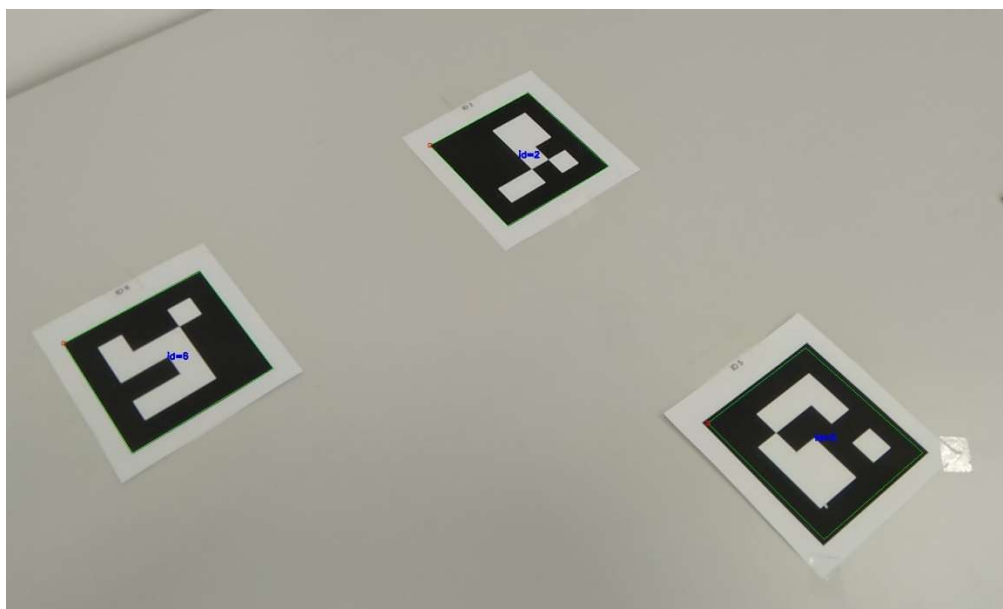
Knihovna Aruco je součástí dodatečných modulů knihovny OpenCV. Je určena pro vytváření a detekci značek, které slouží k určení jejich pozice v obraze. Značka Aruco se skládá ze čtvercové mřížky ohraničené černým okrajem. Výplň mřížky určuje bitová informace (černé políčko je bit s hodnotou 0, bílé bit s hodnotou 1).



Obrázek 11: Značka Aruco 4x4 se znázorněnou mřížkou

Každá značka má svoje vlastní identifikační číslo a jedinečný obrazec. Při detekci značky zjistíme identifikační číslo značky a pozici (pixelové souřadnice) rohů značky. To jsou vlastnosti, které využijeme při zjišťování pozice kamery. Pokud tuto značku umístíme do snímané scény a detekujeme ji, získáme 4 body s jejich pixelovými souřadnicemi. Pokud těmto bodům přiřadíme jejich světové souřadnice, jsme schopni určit rotační a translační vektor. Navíc můžeme použít více různých značek, a každé přiřadit vlastní souřadnice. Při detekci zjistíme identifikační číslo značky a vybereme pro ni ty správné souřadnice. Takto jsme schopni získat dostatek bodů pro zjištění pozice kamery.

Detekce značek Aruco se provádí pomocí funkce `cv::aruco::detectMarkers`, která na daném snímku detekuje značky z předdefinovaného slovníku značek. Výstupními parametry jsou vektor s identifikačními čísly detekovaných značek a vektor s pixelovými souřadnicemi rohů značek. Pokud chceme detekované značky zvýraznit ve snímku, použijeme funkci `cv::aruco::drawDetectedMarkers`.



Obrázek 12: Detekce značek Aruco

Máme tedy soubor bodů v obraze s jejich pixelovými souřadnicemi. Pro tyto body zjistíme jejich světové souřadnice, tedy polohu rohů značek. Nyní musíme z těchto informací vypočítat translační a rotační vektor. K tomu slouží funkce `cv::solvePnP`. Jejimi vstupními parametry jsou vektor s pixelovými souřadnicemi, vektor se světovými souřadnicemi, matice kamery a koeficienty zkreslení, výstupními parametry jsou translační a rotační vektor. Pro sestavení projekční matice potřebujeme vektor rotace převést na matici rotace o velikosti 3×3 . To nám zajistí funkce `cv::Rodrigues`, která má jako vstupní parametr rotační vektor a jako výstupní parametr rotační matici. Nyní již máme všechny potřebné informace k sestavení projekční matice podle rovnice (11).

5.4 Detekce a korespondence bodů

Předtím, než můžeme začít s vlastní rekonstrukcí objektu, musíme na objektu detekovat body, u kterých poté budeme zjišťovat jejich prostorové souřadnice. Výsledný model se tedy bude skládat z detekovaných bodů na objektu, kterým

přiřadíme souřadnice v prostoru. Můžeme tedy říct, že detekce bodů a jejich korespondence je nejdůležitějším bodem této práce, jelikož výsledný model budou představovat pouze tyto body.

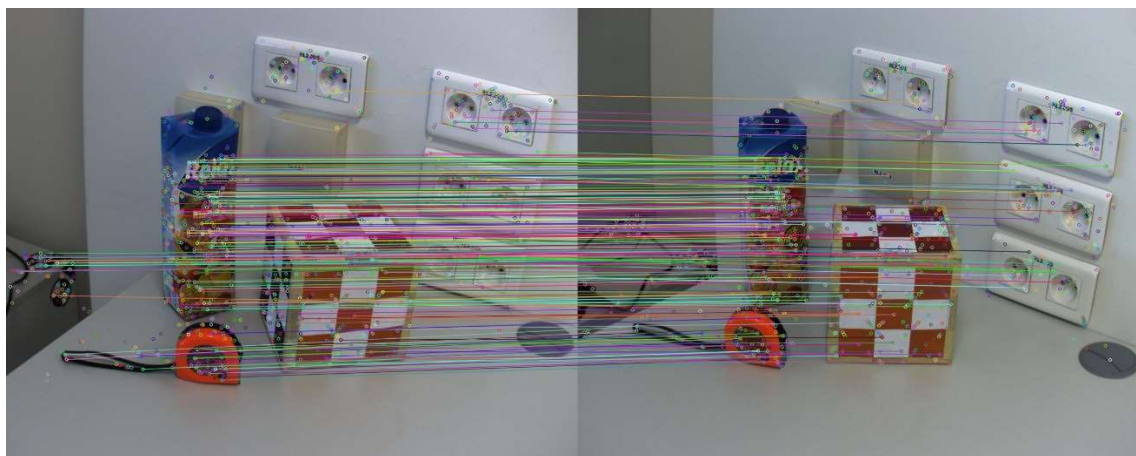
Pro detekci významných bodů bude v této práci použit detektor SURF. Jak je již popsáno výše, detektor SURF je nezávislý na změně měřítka, natočení atd., a rovněž nám spolu s významnými body poskytuje jejich deskriptory. Ty následně využijeme pro nalezení korespondencí. Knihovna obsahující detektor SURF se jmenuje *xfeatures2d* a je součástí dodatečných modulů knihovny OpenCV. Pro vytvoření detektoru použijeme funkci `cv::xfeatures::SURF::create`. Dále použijeme funkci `cv::Feature2D::detectAndCompute`, která nám detekuje významné body a vypočítá jejich deskriptory. Vstupními parametry jsou snímek, na kterém se bude detekce provádět a maska, pomocí které můžeme určit pouze část snímku, ve které budeme body detekovat. Výstupními parametry jsou vektor významných bodů a matice s deskriptory. Pro vykreslení detekovaných významných bodů můžeme použít funkci `cv::drawKeypoints`.



Obrázek 13: Detekce významných bodů pomocí SURF

Dalším důležitým úkolem je nalézt korespondující body mezi levým a pravým snímkem. K tomu využijeme získané deskriptory. Budeme hledat, který deskriptor v druhém snímku je podobný deskriptoru ve snímku prvním. Ke hledání podobných deskriptorů využijeme funkci `cv::DescriptorMatcher::knnMatch`. Ta nám k jedné množině deskriptorů (tzv. „query“) vyhledá nejlepší shodné deskriptory z druhé množiny (tzv. „train“). Dále má tato funkce parametr, který určuje, kolik podobných deskriptorů z druhého snímku bude funkce hledat. Toho se dá využít, pokud chceme omezit falešné korespondence. Pokud budeme hledat dva nejpodobnější deskriptory, můžeme lépe rozhodnout o správnosti nalezené korespondence, a to pomocí rozdílu

deskriptoru z prvního a druhého snímku. Budeme tedy pro každý deskriptor hledat dvojici nejvíce podobných deskriptorů. Pokud bude u jedné nalezené korespondence rozdíl malý a u druhé rozdíl velký, pak můžeme říct, že korespondence s menším rozdílem je ta správná. Pokud bude rozdíl u obou korespondencí přibližně stejný, nemůžeme rozhodnout, která korespondence je ta správná, a tudíž tuto shodu vyřadíme. Stále je tu ale možnost nalezení falešné korespondence. Proto využijeme znalosti epipolární geometrie (viz kapitola 3.1). Ta nám říká, že bod z jednoho snímku leží v druhém snímku na epipolární přímce. Pokud tedy bude bod z nalezených korespondencí ležet příliš daleko od epipolární přímky, můžeme tuto korespondenci považovat za falešnou a vyřadit ji. Pokud předpokládáme, že je správných korespondencí mnohem více než těch falešných, můžeme pro tuto podmínku využít funkci `cv::findFundamentalMat`, která, jak už název napovídá, nalezne fundamentální matici. K tomu využije všech zatím nalezených korespondencí. Jelikož předpokládáme pouze minimální podíl falešných korespondencí, můžeme považovat nalezenou matici za správnou. Kromě fundamentální matice nám tato funkce vypočítá masku, která nám říká, které body leží na epipolární přímce, resp. které body leží do určité vzdálenosti od této přímky (tato vzdálenost je parametr funkce). Tímto způsobem již dokážeme nalézt správné korespondence, resp. omezit falešné korespondence na minimum.



Obrázek 14: Nalezené korespondence

5.5 Výpočet prostorových souřadnic

Jelikož z předchozích kapitol známe postup pro získání pozice kamery a pro získání korespondujících bodů, můžeme nyní přejít k samotnému výpočtu prostorových souřadnic těchto bodů. Budeme využívat princip triangulace, který je popsán v kapitole 3.3. Zrekapitulujme si nyní, co je nutné znát pro použití triangulace. Nejprve potřebujeme znát ohniskovou vzdálenost. Tu zjistíme pomocí kalibrace kamery. Dále potřebujeme znát vzdálenost kamer. K tomu nám poslouží znalost projekční matice. Jelikož nám projekční matice dává informaci o poloze a natočení kamery, nemusíme pomocí triangulace určovat pouze vzdálenost od kamery, ale můžeme rovnou vypočítat prostorové souřadnice bodu. K tomu nám poslouží funkce `cv::triangulatePoints`. Ta po zadání vstupních parametrů, jako jsou vektory s pixelovými souřadnicemi bodů a projekční matice, vypočítá prostorové souřadnice bodů v homogenním tvaru

$[x, y, z, w]$. Abychom dostali nám známé kartézské souřadnice $[X, Y, Z]$, musíme jednotlivé souřadnice x, y, z vydělit poslední souřadnicí w , tedy:

$$X = \frac{x}{w}, Y = \frac{y}{w}, Z = \frac{z}{w}, w \neq 0 \quad (20)$$

5.6 Vizualizace získaných dat

Posledním úkolem při prostorové rekonstrukci objektu bude vizualizace získaných dat, tedy zobrazení získaných prostorových bodů v grafické podobě. Jelikož zobrazení pouhých bodů může být často velmi nenázorné, je vhodné kromě polohy bodu zobrazit i jeho barvu, kterou získáme během procesu detekce bodu. Proto byl pro uložení a zobrazení dat vybrán grafický formát VRML.

VRML (Virtual Reality Modeling Language) je grafický formát pro popis trojrozměrné scény. Zobrazovaná data jsou zapsána v textové formě, jako například program napsaný v programovacím jazyce. Formát VRML slouží k popisu libovolných těles v prostoru, včetně vlastností jeho povrchu (barva, průhlednost, materiál) a vlastností okolí (světlo). Soubor VRML má příponu `.wrl` a je textový, takže ho lze upravit běžnými textovými editory.[10]

Pro účely této práce bude stačit pouze zobrazení bodů a jejich barev. Ty budeme zobrazovat tak, že pro každý bod vytvoříme sféru s malým průměrem, jejíž střed bude v souřadnicích daného bodu. Takto bude vypadat zápis jednoho bodu v jazyce VRML:

```
#VRML V2.0 utf8

Transform {
  translation 0.123 -0.100 0.065  #souradnice bodu
  children[
    Shape{
      appearance Appearance{
        material Material {
          diffuseColor 0.69 0.69 0.63  #barva v RGB
        }
      }
      geometry Sphere {
        radius 0.003  #prumer sfery
      }
    }
  ]
}
```

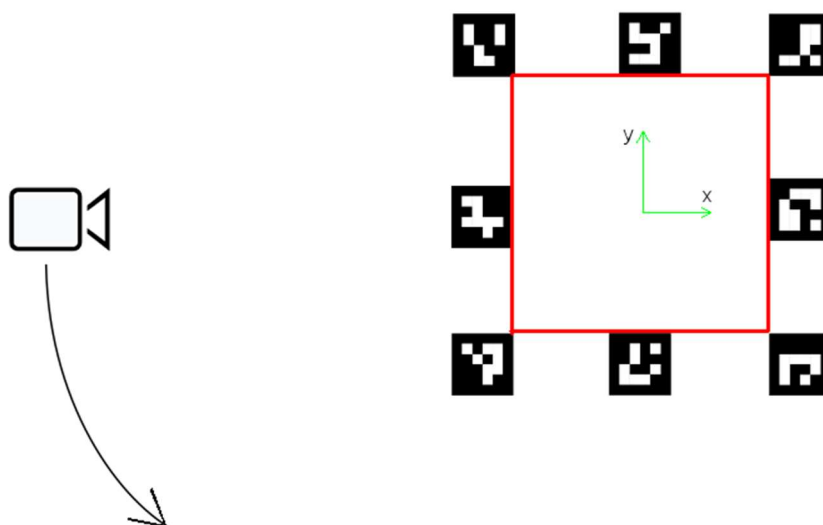
Pro zobrazení souboru VRML můžeme využít např. program Cortona3D Viewer, který nám soubory s příponou `.wrl` otevře ve webovém prohlížeči.

5.7 Měřicí pracoviště

Měřicí pracoviště pro tuto práci se bude skládat z prostoru pro umístění objektu a kamery, která bude danou scénu s objektem snímat. Byla zvolena koncepce s jednou

pohyblivou kamerou. To znamená, že snímáný objekt umístíme na plochu, a kamerou budeme objekt postupně snímat z různých úhlů tak, abychom kamerou postupně nasníмали objekt ze všech stran. Jelikož budeme používat pouze jednu kameru, budou se výsledné stereo snímky nutné pro rekonstrukci objektu skládat ze dvou sousedních snímků téže kamery.

Plocha pro umístění objektu bude obsahovat značky Aruco, které využijeme pro zjištění pozice kamery (viz kapitola 5.3). Ty musí být umístěny tak, aby ohraničovaly prostor pro objekt. Celé pracoviště si dále popíšeme pomocí obrázku č. 15.



Obrázek 15: Návrh měřicího pracoviště

Celkem bude pracoviště obsahovat 8 různých značek Aruco, které pravidelně rozmístíme okolo plochy vyhrazené pro snímáný objekt, která je na obrázku znázorněna červeně. Velikost této plochy je 40x40 cm, což nám udává maximální rozměry snímaného objektu. Pokud by byl objekt větší, překrýval by některé značky a hrozilo by, že nebude možné zjistit pozici kamery v prostoru. Dále je na obrázku znázorněn světový souřadný systém (zeleně). Jeho počátek je umístěn ve středu plochy pro objekt a osa z směřuje směrem vzhůru. Umístění kamery musíme zvolit tak, aby její zorné pole zabíralo celý objekt včetně okolních značek. Pro pořízení dalšího snímku kameru přemístíme po pomyslné kružnici kolem objektu proti směru hodinových ručiček (na obrázku naznačeno šipkou). Takto s kamerou postupujeme okolo objektu, abychom ho nasníмали ze všech stran.



Obrázek 16: Měřicí pracoviště

5.8 Popis programů

V rámci této práce byly vytvořeny dva programy. Jeden program slouží pro kalibraci kamery a druhý pro vytvoření 3D modelu daného objektu. Programy byly vytvářeny v jazyku C++ ve vývojovém prostředí Microsoft Visual Studio 2017. V následujících dvou podkapitolách si přesně popíšeme tyto dva programy.

5.8.1 Program pro kalibraci kamery – CameraCalibration

Tento program slouží pro kalibraci kamery, tedy pro zjištění vnitřních parametrů kamery a koeficientů zkreslení. Využívá funkce popsané v kapitole 5.2.

Program nejprve načte kalibrační snímky ve formátu JPG. Ty musí být pojmenovány c0.jpg až cx.jpg, kde x je index posledního snímku, a musí být uloženy ve složce „CalibImages“, která je umístěna ve složce společně s programem. Program vyžaduje nejméně 15 těchto snímků. Snímky se načítají ve smyčce a po každém načteném snímku se ověří, zda obsahuje šachovnicový vzor. Pokud ano, snímek se přidá do vektoru snímků ke kalibraci a pokud ne, vyřadí se a vypíše se hláška „Chessboard not found!“. Po načtení všech snímků se vypíše hláška „Loaded n images“, kde n je počet načtených snímků.

Program pokračuje tím, že u jednotlivých snímků detekuje rohy šachovnicového vzoru a uloží do vektoru jejich pixelové souřadnice. Těmto rohům poté přiřadíme světové souřadnice tak, že levý horní roh má souřadnice $[0, 0, 0]$ a souřadnice dalších se vypočítají díky znalosti rozměru pole šachovnice. Tyto souřadnice rovněž uložíme do vektoru. Poté se tyto dva vektory, tj. vektor s pixelovými souřadnicemi rohů šachovnice a vektor s jejich světovými souřadnicemi, využijeme pro kalibraci. Tím zjistíme matici vnitřních parametrů kamery a koeficienty zkreslení. Po výpočtu parametrů program

vypíše hlášku „Camera calibrated“. Získané informace program uloží do souboru ve formátu XML. Tento soubor využívá další program pro sestavení 3D modelu.

5.8.2 Program pro vytvoření modelu – 3DModel

Program „3DModel“ slouží k získání prostorového modelu snímaného objektu. Snímaný objekt musí být umístěn na vyhrazeném místě v měřicím pracovišti. Popíšeme si nyní jednotlivé body tohoto programu. Při tom využijeme funkcí popsaných v kapitolách 5.3, 5.4 a 5.5.

Po spuštění programu se nejprve načte kalibrační soubor, který jsme získali pomocí programu CameraCalibration. Tento soubor obsahuje informace nutné k sestavení projekční matice, jako matici kamery a koeficienty zkreslení. Musí být umístěný ve složce společně se zdrojovými soubory. Pokud soubor neexistuje nebo nejde načíst, vypíše se chybová hláška a program se ukončí. Pokud načtení kalibračního souboru proběhlo v pořádku, začne program načítat snímky. Ty jsou umístěné ve složce „ScanImages“, která se nachází ve složce se zdrojovými soubory. Ty musí být pojmenovány s0.jpg až sx.jpg, kde x je index posledního snímku. Pokud program žádné snímky nenačte, vypíše chybovou hlášku a ukončí se.

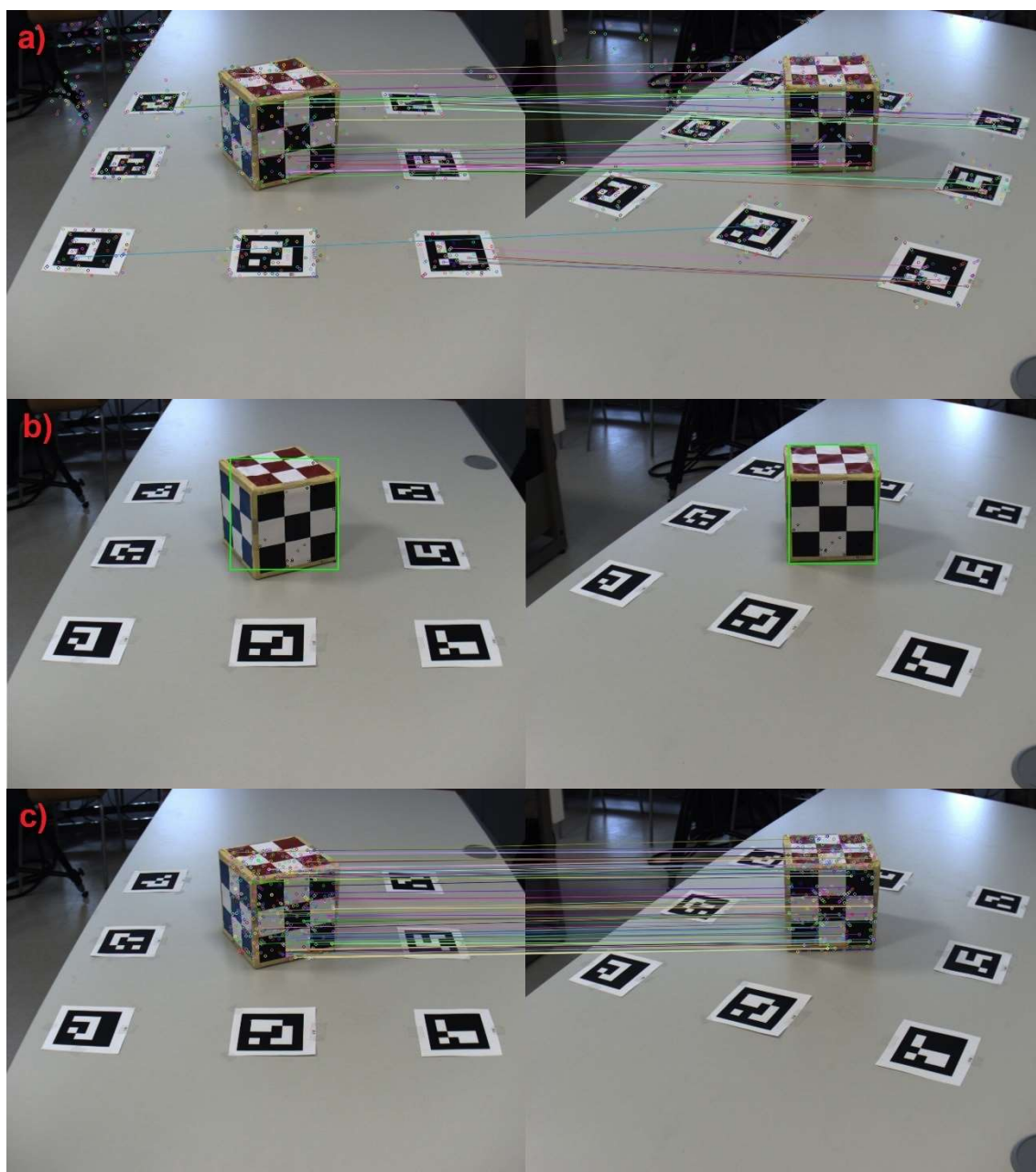
Program pracuje v cyklech. V každém cyklu načte dvojici snímků a provede procesy vedoucí k vytvoření části modelu snímaného objektu. Počet cyklů závisí na počtu snímků objektu.

V prvním cyklu program načte první dvojici snímků, tedy s0.jpg a s1.jpg. Ty nám budou představovat stereo dvojici snímků, tedy levý a pravý snímek. Nejprve se ověří, zda je na snímcích detekována alespoň jedna značka Aruco. Pokud se značky nedetekují, program pokračuje dalším cyklem, tzn. načítá další dvojici snímků. Pokud jsou značky detekovány, zjistí se pixelové souřadnice jejich rohů. K těm je nutné přiřadit světové souřadnice. Ty jsou v programu pro každou značku zadány. Podle identifikačních čísel se tedy jednotlivým pixelovým souřadnicím přiřadí souřadnice světové. To jsou informace, podle kterých můžeme sestavit projekční matici kamery pro každý snímek. K tomu ještě využijeme informace z kalibračního souboru, tedy matici kamery a koeficienty zkreslení. Využívá se funkcí popsaných v kapitole 5.3 a znalostí z kapitoly 2.5. Máme tedy pro levý i pravý snímek sestavenou projekční matici.

Dalším úkolem programu je detekce významných bodů na snímcích a jejich korespondence. Postup a využití funkce jsou popsány v kapitole 5.4. Kromě toho se zjistí i barvy jednotlivých bodů a uloží se do vektoru. Toho můžeme využít při vykreslování modelu a prostorovým souřadnicím tuto barvu přidělit, model pak lépe vystihuje skutečný objekt. K výpočtu prostorových souřadnic z nalezených korespondencí využijeme postup uvedený v kapitole 5.5.

Jelikož významné body detekujeme na celém snímku, dochází k tomu, že v zejména v okrajích snímku a okrajích měřicího pracoviště detekujeme mnoho navzájem podobných bodů. To značně ztěžuje nalezení správných korespondencí. Proto je nutné zpřísnit kritéria pro posuzování správnosti korespondencí. To ovšem znamená, že jejich výsledný počet bude nízký, a tudíž se bude výsledný model skládat pouze z malého množství bodů. Tomuto problému se dá zabránit tím, že budeme významné body hledat pouze v určité oblasti snímku, a to tam, kde se nachází snímaný objekt. Zbývá otázka, jak tuto oblast určit. Toho docílíme tak, že po výpočtu prostorových souřadnic nalezených korespondencí vyřadíme ty, které se nacházejí mimo prostor vyhrazený na snímaný objekt. Zbydou nám body, které byly detekovány na snímaném objektu.

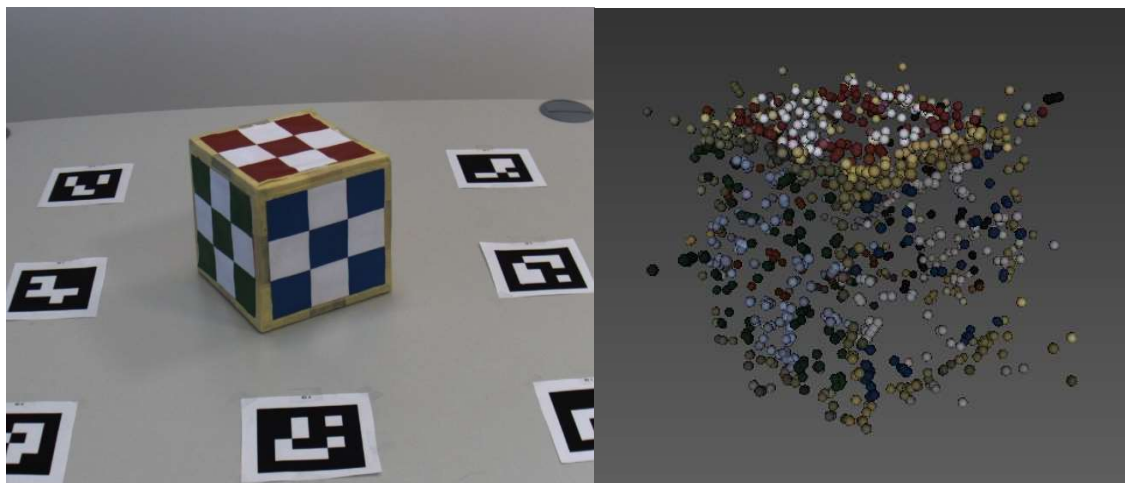
Zpětně zjistíme pixelové souřadnice těchto bodů, a okolo těchto bodů sestavíme obdélník. Ten nám bude vymezovat oblast, kde budeme opět hledat korespondence, nyní už ovšem s méně přísnými kritérii pro posouzení jejich správnosti, jelikož na objektu se nachází mnoho navzájem odlišných bodů. Je možné, že tato oblast nebude přesně obepínat celý objekt, ale pokud bude oblast zabírat alespoň část objektu, dokážeme i na této části detekovat více bodů pro rekonstrukci. Poté celý proces detekce, korespondence a výpočtu prostorových souřadnic opakujeme. Tím zajistíme dostatečný počet bodů pro výsledný model.



Obrázek 17: Postup hledání korespondencí

- a) korespondence na celém snímku; b) oblast, kde se nachází objekt;
c) korespondence v oblasti objektu

Výše uvedený postup program opakuje v cyklech, dokud postupně nezpracuje všechny snímky. Po zpracování poslední dvojice snímků program uloží nalezené prostorové body. Ty bude postupně zapisovat do souboru v jazyce VRML (viz kapitola 5.6). Po zpracování všech bodů program uloží soubor „3DModel.wrl“, vypíše hlášení o uložení souboru a poté se ukončí.



Obrázek 18: Ukázka modelu krychle

6 ZÁVĚR

Cílem této bakalářské práce bylo nastudovat metody trojrozměrné rekonstrukce a sestavit trojrozměrný model objektu pomocí více pohledů kamer.

V první části práce byla rozebrána problematika kamer a jejich matematický popis, technika pohledů dvou kamer a v neposlední řadě metody pro detekci významných bodů v obraze. Tyto teoretické znalosti byly použity pro návrh postupu při rekonstrukci objektu.

Samotný postup při tvorbě modelu je rozebrán v druhé části práce. Byl popsán způsob kalibrace kamery, detekce a korespondence bodů a výpočtu prostorových souřadnic bodů. To vše pomocí knihovny OpenCV. Je zde také popsán způsob vykreslování modelu.

V rámci této práce vzniklo měřicí pracoviště. To využívá koncepce jedné pohyblivé kamery, takže páry snímků vznikají při snímání objektu z různých pozic. Byl navržen postup získávání informace o pozici kamery pomocí rozmístěných značek Aruco. Dále vznikly v rámci práce programy, v nichž jsou jednotlivé metody implementovány. Výsledky této práce si můžeme prohlédnout v příloze.

Největším problémem při tvorbě modelu byla bezpochyby detekce a korespondence bodů. Prvním problémem je nízký počet detekovaných bodů. To může být způsobeno snímáním málo texturovaného objektu. Tento problém se dá vyřešit pouze úpravou objektu, například dodatečným polepením texturou či speciálním nasvícením objektu, např. laserovým vzorem. Druhým problémem jsou falešné korespondence, které poté způsobují vykreslování bodů, které zjevně nejsou součástí objektu, a výsledný model znehodnocují.

Cílem další práce by tedy mohlo být zdokonalení techniky hledání korespondencí, nalezení jiné metody hledání pozice kamery nebo využívání technik nasvícení objektu.

Literatura

- [1] Žára J., Beneš B., Sochor J., Felkel P.: Moderní počítačová grafika, Computer Press, 2005, ISBN 978-80-2-1-0454-5
- [2] OpenCV [online]. © 2019, OpenCV team. [cit. 2019-04-24]. Dostupné z: <https://opencv.org/about/>
- [3] OpenCV documentation [online]. ©2011-2014, opencv dev team. [cit. 2019-04-24]. Dostupné z: <https://docs.opencv.org/2.4/index.html>
- [4] KHAN, Nazar. *Computer Vision: Camera Geometry* [online]. [cit. 2019-03-06]. Dostupné z: http://faculty.pucit.edu.pk/nazarkhan/teaching/Fall2018/CS465/Lectures/lecture11_camera_geometry.pdf.
- [5] Dissecting the Camera Matrix, Part 2: The Extrinsic Matrix. Sightations [online]. 2012 [cit. 2019-03-06]. Dostupné z: <http://ksimek.github.io/2012/08/22/extrinsic/>
- [6] Slovníček pojmů machine vision. *Analýza obrazu* [online]. 2019 [cit. 2019-03-06]. Dostupné z: <http://www.analyza-obrazu.cz/strojove-videni/slovnicek-pojmu/>
- [7] Epipolar Geometry and the Fundamental Matrix. *Information Engineering* [online]. [cit. 2019-03-13]. Dostupné z: <http://www.robots.ox.ac.uk/~vgg/hzbook/hzbook2/HZepipolar.pdf>
- [8] HORÁK, Karel. *Lokální příznaky a korespondence* [online]. [cit. 2019-03-22]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/Lectures/03_Lokalni_priznaky_a_korespondence.pdf
- [9] HORÁK, Karel. *Deskriptory oblastí* [online]. [cit. 2019-03-22]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ROZ/Lectures/05_Deskriptory_oblasti.pdf
- [10] *Výukové pásmo VRML* [online]. [cit. 2019-05-13]. Dostupné z: <http://dcegi.fel.cvut.cz/home/zara/vyuka/vrml/tutorial/pasmo/>

Seznam příloh

Příloha 1	Obsah CD	34
Příloha 2	Ukázka modelů.....	35

Příloha 1 Obsah CD

Na přiloženém CD naleznete:

- Bakalářskou práci ve formátu PDF
- Kompletní zdrojové kódy programů vytvořených v rámci práce
- Databázi kalibračních snímků
- Databázi měřících snímků
- Výsledné modely ve formátu VRML

Příloha 2 Ukázka modelů

Všechny uvedené modely jsou zobrazeny v prohlížeči Cortona3D Wiewer a jsou dostupné na přiloženém CD ve formátu VRML.

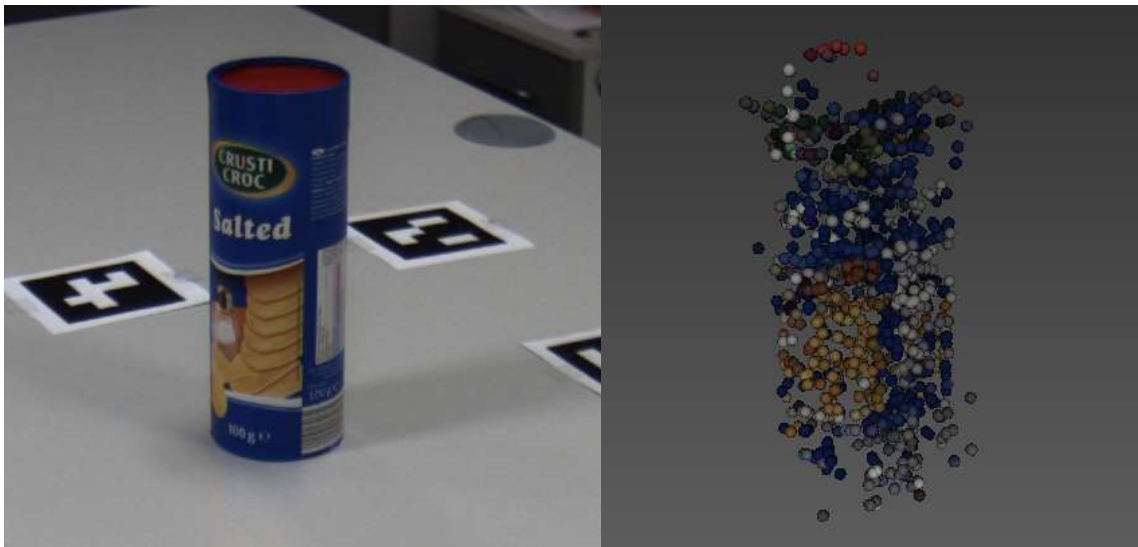
Model č. 1



Model č. 2



Model č. 3



Model č. 4



Model č. 5



Model č. 6

- Zde je vidět důsledek nedostatečného počtu detekovaných bodů – model je nenázorný a nepoužitelný

